Découvrons la RIPE database

Posté le 18 avril 2014 à 23h01 par GuiGui

Le RIPE, comme tout RIR, dispose d'une base de données publique qui regroupe toutes (un maximum on va dire) les informations "technico-administratives" (à qui a été alloué telle ressource numérique unique ? qui est le responsable technique de ce réseau, quel AS doit annoncer tel préfixe IP ?, ...) concernant les réseaux informatiques publics en Europe (version large) et au Moyen-Orient.

Habituellement, on accède à cette base de données lorsque l'on fait une requête avec le protocole whois sur une ressource (plage d'IP, numéro d'AS, ...) allouée/assignée par le RIPE. Aujourd'hui, on va plutôt regarder de quoi est constituée cette base de données et comment les LIR et les opérateurs réseaux la mettent à jour (ou non, d'ailleurs).

Si tous les RIR ont une base de données, pourquoi je me focalise sur celle du RIPE ? Simplement car le RIPE est bien de chez nous et comme le format de la base (RPSL ou non, objets/attributs différents, ...) change entre RIR, autant étudier ce qui nous concerne. De plus, la RIPE DB est la mieux documentée, à mon humble avis.

Attention : ce billet n'est ni de la vulgarisation ni une source d'information fiable. Il s'agit juste d'un retour d'expérience et de quelques notes d'un n00b qui s'est penché sur le sujet.

Table des matières

- Table des matières
- De la documentation générale
- Documentation spécifique
- Quelques paramètres whois bien utiles
- RIPE TEST Database
  - person/mntner
  - organisation
  - inetnum/inet6num
  - ASN
  - Délégation zone reverse
  - Politiques de routage inter-AS
  - Cogestion
  - Modifier la base par mail avec signature PGP
  - Faire le ménage dans la base de test
  - Dernières remarques

De la documentation générale

- Protocole whois défini dans le RFC 3912.
- Routing Policy Specification Language documenté dans les RFC RFC 2622 - 2725 - 4012

(mais je ne pense pas qu'une lecture complète de ces RFC soit pertinente).

Documentation spécifique

- La documentation de la RIPE database et particulièrement :
  - RIPE Database User Manual: Getting Started
  - RIPE Database Update Reference Manual

Archives :

avril 2014

L	Ma	Me	J	V	S	D
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

« mar mai »

Catégories

- Administration réseau
- Administration système
- BGP & RPKI+ROA
- Blog
- Cinéma/TV
- Développement
  - Assembleur 8086+
- Divers
- DNS & DNSSEC
- Hardware
  - Embarqué
    - OpenWRT
- Humeur
- Jeux vidéo
- Logiciels
- Sécurité

- Plus orienté sur la pratique et à destination des FAI associatifs qui veulent modifier directement les objets créés par leur LIR dans la base du RIPE : Maintenance Whois chez Lulu

Quelques paramètres whois bien utiles

Ce que je retiens principalement des lectures précédentes, c'est des paramètres de la commande whois, qui sont spécifiques à certaines bases comme celle du RIPE mais qui n'en sont pas moins très utiles :

- whois -t <type_objet> : afficher les attributs d'un type d'objet
- whois -B : afficher toutes les infos (mail ...) de l'objet demandé (sans ça : « source: RIPE # Filtered »)
- whois -L : remonter toute l'arborescence d'une allocation. Exemple pour un inetnum : organisation - LIR - RIR - IANA
- whois -l : pareil que précédemment mais remonte uniquement d'un étage/d'une étape/d'un cran
- whois -r -i org <id_organisation> : afficher toutes les ressources/objets assignés/alloués à une organisation (objet de type « organisation »)

RIPE TEST Database

Le RIPE met à disposition une base de données de test en libre-service pour que chacun puisse découvrir/s'entraîner à l'arrache. Je pense que c'est un excellent moyen de voir comment ça se passe de l'autre côté du whois. C'est vraiment intéressant et formateur.

Ci-dessous, je ne vais pas être exhaustif, juste donner des pistes et énoncer quelques "pièges" que j'ai rencontrés pour vous éviter de tomber dedans.

Dans mes exemples, je serais Jean-Kevin Boulay de l'association TESTASSO.

person/mntner

La première étape est de créer un objet de type « person ». Voyons les différents attributs disponibles :

```
whois -t person
person:         [mandatory]  [single]     [lookup key]
address:        [mandatory]  [multiple]   [ ]
phone:          [mandatory]  [multiple]   [ ]
fax-no:         [optional]   [multiple]   [ ]
e-mail:         [optional]   [multiple]   [lookup key]
org:            [optional]   [multiple]   [inverse key]
nic-hdl:        [mandatory]  [single]     [primary/lookup key]
remarks:        [optional]   [multiple]   [ ]
notify:         [optional]   [multiple]   [inverse key]
abuse-mailbox:  [optional]   [multiple]   [inverse key]
mnt-by:         [mandatory]  [multiple]   [inverse key]
changed:        [mandatory]  [multiple]   [ ]
source:         [mandatory]  [single]     [ ]
```

Hum ... mnt-by doit pointer un objet de type « mntner » (maintainer) ... Quels sont ses attributs ?

```
whois -t mntner
mntner:         [mandatory]  [single]     [primary/lookup key]
descr:          [mandatory]  [multiple]   [ ]
org:            [optional]   [multiple]   [inverse key]
admin-c:        [mandatory]  [multiple]   [inverse key]
tech-c:         [optional]   [multiple]   [inverse key]
upd-to:         [mandatory]  [multiple]   [inverse key]
mnt-nfy:        [optional]   [multiple]   [inverse key]
auth:           [mandatory]  [multiple]   [inverse key]
remarks:        [optional]   [multiple]   [ ]
notify:         [optional]   [multiple]   [inverse key]
abuse-mailbox:  [optional]   [multiple]   [inverse key]
mnt-by:         [mandatory]  [multiple]   [inverse key]
referral-by:    [mandatory]  [single]     [ ]
changed:        [mandatory]  [multiple]   [ ]
source:         [mandatory]  [single]     [ ]
```

Il y a une référence circulaire : un objet person dépend d'un objet de type maintainer mais un objet maintainer dépend aussi d'un objet de type person (attribut « admin-c »).

La solution ? Utiliser le formulaire pour les nouveaux arrivants dans la RIPE Database.

Mode d'emploi rapide :

- Source : bien choisir la TEST Database
- Person : prénom nom. Exemple : J-K Boulay
- NIC hdl : ne peut pas être vide contrairement à ce qui est indiqué. Il s'agit de la clé primaire de l'objet. Elle doit être unique dans la base donc la base l'attribue elle-même. Vous pouvez mettre :
  - Soit une suite de caractères qui respecte le formalisme (« From 2 to 4 characters optionally followed by up to 6 digits optionally followed by a source specification. The first digit must not be "0". Source specification starts with "-" followed by source name up to 9-character length. »). Exemple : JKB1-TEST. Attention : un nic-hdl déjà pris générera une erreur « The nic-hdl "JKB1-TEST" is not available ».
  - Soit la valeur « AUTO-1 » pour laisser complètement le choix à la base ou « AUTO-1<abréviation> ». Abréviation est composée de 1 à 4 lettres. Exemple : AUTO-1JKB pour obtenir le nic-hdl : « JKB1-TEST ». Il est inutile d'essayer de mettre AUTO-42JKB pour tenter d'obtenir un nic-hdl « JKB42-TEST » : ça ne marche pas. Je n'ai pas trouvé comment forcer le nombre et c'est à mon avis bien normal : ID, on prend le suivant, même si un objet avec un ID inférieur a été supprimé depuis, tout ça.
- Address : FR suffit, par exemple
- Phone : +33123456789 suffit, par exemple
- Email : ripedb@example.com suffit, par exemple
- Pour le mntner, le nom est une clé et on reprend souvent le nom de l'organisation à laquelle la personne qui s'inscrit est rattachée. Exemple : Maintainer name : TESTASSO-MNT

Normalement, vous obtenez le message suivant :

```
We have created two RIPE Database objects for you, please make a note of the primary ke

person nic-hdl: JKB1-TEST

person:         J-K Boulay
address:        FR
phone:          +33123456789
e-mail:         ripedb@example.com
nic-hdl:        JKB1-TEST
mnt-by:         TESTASSO-MNT
changed:        ripedb@example.com 20140418
source:         TEST

maintainer name: TESTASSO-MNT

mntner:         TESTASSO-MNT
descr:          Startup maintainer
admin-c:        JKB1-TEST
upd-to:         ripedb@example.com
auth:           MD5-PW blabla
mnt-by:         TESTASSO-MNT
referral-by:    TESTASSO-MNT
changed:        ripedb@example.com
source:         TEST
```

Petite note : un objet de type maintainer est un verrou qui empêche la modification et la suppression inopinée de vos enregistrements présents dans la base. Il recense également les moyens qui permettent d'authentifier le requérant d'une modification/suppression. Actuellement, la base supporte une auth par mot de passe (stocké sous forme de hash MD5 dans la base) ou par signature PGP des objets (uniquement par

mail, pas par l'interface web. Oui, on peut créer/modifier des objets par mail.). Pour l'instant, nous n'avons qu'une auth par mot de passe.

Vous pouvez effectuer une recherche pour vérifier que vos objets ont bien été créés : https://apps.db.ripe.net/search/query.html.

Si vous souhaitez modifier vos objets, c'est par là : https://apps.db.ripe.net/webupdates/search.html. Cependant, lors de la soumission de vos modifications, vous allez recevoir l'erreur : « No authorisation has been supplied ». En effet, vous n'avez pas prouvé votre identité selon l'un des mécanismes prévus dans l'objet maintainer qui verrouille l'objet que vous souhaitez modifier. Pour vous authentifier, il suffit de saisir votre mot de passe de maintainer dans la zone « Session Passwords » dans le menu de droite.

À partir de maintenant, vous pouvez créer vos objets à l'adresse suivante : https://apps.db.ripe.net/webupdates/select-type.html. Deux méthodes de saisies sont disponibles : format brut (que du texte) ou formulaire avec uniquement les attributs obligatoires.

organisation

Un objet de type « organisation » permet de regrouper toutes les ressources associées/allouées à une organisation donnée. Très pratique mais pas obligatoire. Les autres objets peuvent pointer dessus avec un attribut « org ». L'attribut organisation fonctionne comme le NIC-hdl d'une person donc « AUTO-1<abréviation> ».

Exemple :

```
organisation:  ORG-TSTA1-TEST
org-name:  Association TESTASSO
org-type:  OTHER
address:  FR
e-mail:  ripedb@example.com
mnt-ref:  TESTASSO-MNT
mnt-by:  TESTASSO-MNT
changed:  ripedb@example.com 20140418
source:  TEST
```

inetnum/inet6num

Passons au plus intéressant : une plage d'adresses IPv4 : un inetnum (ou v6 avec inet6num, ça fonctionne pareil).

En situation réelle, vous n'aurez pas besoin de créer cet objet : c'est le boulot de votre LIR qui vous mettra en mnt-* si vous le souhaitez pour que vous ayez des droits sur vos allocations. De plus, en situation réelle, le processus d'allocation de ressources ne se passe pas aussi facilement en mode "ho, cette plage est disponible, je la prends". Néanmoins, ici, je choisis 192.0.2.0/24 qui est une plage réservée à la documentation.

Ici, il faudra l'accord de la hiérarchie, c'est à dire de l'organisation (RIR ou LIR) a qui a été attribuée la plage d'adresses IP qui englobe celle que cette organisation vous a attribuée. Pour trouver quel est le maintainer responsable, il suffit de faire une recherche.

Via l'interface web :

- Search term : 192.0.2.0 - 192.0.2.255
- Sources : TEST Database
- Hierarchy Flags : l (l comme lapin)

Ou via whois : whois -h whois-test.ripe.net -l 192.0.2.0 - 192.0.2.255

Voilà un extrait de la réponse :

```
inetnum:        0.0.0.0 - 255.255.255.255
netname:        IANA-BLK
descr:          The whole IPv4 address space
country:        EU # Country is really world wide
org:            ORG-TT1-TEST
admin-c:        AA1-TEST
tech-c:         AA2-TEST
status:         ALLOCATED UNSPECIFIED
remarks:        The country is really worldwide.
```

```
mnt-by:          TEST-ROOT-MNT
mnt-lower:       TEST-DBM-MNT
mnt-routes:      TEST-DBM-MNT
remarks:         This is an automatically created object.
source:          TEST # Filtered
```

On regarde l'attribut « mnt-lower: ». Le maintainer est « TEST-DBM-MNT ». Dans cette base de test, le mot de passe est donné dans un attribut « remark » de cet objet mntner. Je vous laisse le retrouver et l'ajouter à vos « Session Passwords » (menu de droite, tout ça ...).

Maintenant, vous pouvez enregistrer un inetnum. Exemple :

```
inetnum:  192.0.2.0 - 192.0.2.255
netname:  TESTASSO-MAIN
descr:  IPv4 TESTASSO allocation
country:  FR
admin-c:  JKB1-TEST
tech-c:  JKB1-TEST
status:  ASSIGNED PA
mnt-by:  TESTASSO-MNT
changed:  ripedb@example.com 20140418
source:  TEST
```

Remarquez que le statut est fantaisiste et ne correspond pas à la réalité : l'IANA n'alloue pas directement des ressources. Le RIPE le faisait (PI + allocation directe). Mais c'est fini (famine IPv4, tout ça et il me semble que c'est pareil en IPv6). Donc, au minimum, il faudrait un RIR et un LIR entre TESTASSO et l'IANA avec autant d'objets. Si ça vous tente de faire une maquette plus réaliste ... Ce n'est pas difficile, juste répétitif.

ÉDIT du 26/04/2014 à 22h00 : Remarquez que le mnt-by est lui aussi fantaisiste : un LIR ne vous mettra pas en mnt-by car il s'agit de ses allocations à lui en tant que LIR tout comme le RIPE n'a pas défini le LIR comme mnt-by sur le(s) inet(6)num qu'il a attribué au LIR. C'est le rôle de ces organismes (RIR et LIR) de délimiter les allocations. Sur les inet(6)num, votre LIR vous mettra en mnt-lower pour gérer des découpages dans vos allocations (exemple : fourre-tout v6), en mnt-domain pour gérer les objets domain associés et en mnt-routes pour créer/modifier les objets de type route, ... **Fin de l'édit**

ASN

Pour un numéro de système autonome (ASN), le type d'objet est « aut-num ». Sa création ne pose pas de problèmes si vous êtes arrivé jusque-là.

Délégation zone reverse

Pour la délégation d'une zone reverse v4/v6, c'est un objet de type « domain » qu'il vous faut. Mais dans la base de test, sa création échouera car le niveau supérieur (0/0 en v4) n'a pas configuré les serveurs de noms qui font autorité sur cette zone reverse (et pour cause ...).

Politiques de routage inter-AS

Pour ce qui est des politiques de routage, on a les objets de type route/route6 qui permettent de dire : « tel préfixe IPv4 ou IPv6 est annoncé par tel AS » ou des choses plus marrantes comme « quelle est l'adresse IP de la machine que ce réseau met à disposition du public pour servir d'amer ? ».

Pour les politiques de routage, on a aussi d'autres objets (filter-set, par exemple) et des attributs dans l'objet « aut-num » : import, export et les types liés (as-set, route-set). Tout cela constitue l'Internet Routing Registry (IRR). Exemple :

```
aut-num: AS197422
as-name: TETANEUTRAL-NET-AS
[...]
from AS31576 accept ANY
from AS6777 accept ANY
from AS6939 accept ANY
[...]
to AS31576 announce AS197422
to AS6939 announce AS197422
to AS6777 announce AS197422
[...]
```

Ici, ce FAI toulousain annonce que ses pairs sont GIXE (transitaire), l'AMS-IX (point d'échange néerlandais) et Hurricane Electric (gros opérateur). Tetaneutral.net récupère toutes les routes auprès d'eux sans prioriser un pair vis-à-vis d'un autre. Enfin, Tetaneutral annonce ses préfixes à ses pairs.

Cogestion

Et pour faire de la cession de droits ou, à minima, de la cogestion sur des objets ? Il suffit de leur associer de nouveaux objets de type mntner dans leurs attributs « mnt-by » ou « mnt-* » pour des droits plus fins. Exemples : mnt-routes pointe sur un objet maintainer qui aura le droit d'ajouter des objets de type route/route6 en rapport avec l'objet courant ; mnt-domain = même principe mais pour les reverse v4/v6. L'authentification fonctionne selon un OU logique : toute personne qui aura un des mécanismes d'authentification d'un des maintainers pourra modifier l'objet.

ÉDIT du 26/04/2014 à 23h55 :

Modifier la base par mail avec signature PGP

Documentations : PGP Authentication in the RIPE Database — RIPE Network Coordination Centre ainsi que le tutoriel « Maintenance whois de Lulu » que j'ai linké en début de billet.

Notez qu'il n'est pas nécessaire d'utiliser PGP : vous pouvez envoyer vos modifications par mail en y ajoutant un attribut « password » contenant le mot de passe de votre objet maintainer en clair. Mais c'est mal, très mal. La réciproque n'est pas vraie : l'interface web ne supporte pas les signatures PGP.

Les robots à contacter (source) :

- auto-dbm(at)ripe(dot)net pour la vraie DB
- test-dbm(at)ripe(dot)net pour la DB de test (un greylisting est en place sur cette adresse)

La première étape pour soumettre vos modifications signées avec PGP par mail est de créer un objet de type « key-cert ». Il semble que le formulaire « individual fields » est cassé (il retourne toujours l'erreur « certif value is required ») donc utilisez le formulaire « single text area ».

KEY-CERT : PGPKEY-<ID_de_votre_cle>

CERTIF : votre clé publique au format ASCII obtenu avec gpg --armor --export <ID_de_votre_cle>. Il faut autant de lignes « certif » que de lignes dans la sortie gpg. Il faut conserver les lignes vides et les marqueurs de début et de fin de la clé.

Les autres attributs sont soit générés automatiquement (« method », « owner », « fingerpr ») soit facultatifs. On supprime les premiers. On complète ou non les seconds.

Exemple :

```
KEY-CERT:         PGPKEY-698DB1DD
CERTIF:           -----BEGIN PGP PUBLIC KEY BLOCK-----
CERTIF:           Version: GnuPG v1.4.12 (GNU/Linux)
CERTIF:
CERTIF:           mQINBEvXHzcBEACwopvTtQLlh+vfqwolOHaplhV8x1zFjd1lT1E0CLwpsC5qpmQb
CERTIF:           o5bNlSoDDqlMG6IHP98TUWM24mZU626MuWHte/vMOE5X1Q7tk0TRREeYjXxBftaa
CERTIF:           9pGhynrjdk4FSDSZjZ5N0CgCDeX61j1TgCH0LT1EPB0/hJ+Tv2jDHc0vfCq6tnCg
CERTIF:           a3GoKrAlPOWff/nNFpXQwpLbeFDizurGXNDpGa5yBPfn0k8bKzhPa+h9geYIx+4e
CERTIF:           NbtG0J/g+GJuLhmlZyMVQ5+VYgeggbwLw8CvvcsH82v5PG5XIyceMqEXOvji5BOL
CERTIF:           jzuYFYF/XinPbWgz5q2+5hV3HsJz8jMHlkPZnj1hbHa9mT1A+xSFUQ95gbvRXgF2
CERTIF:           iOmewYUVPKoOfIhGis1cq1f4SlhrgO0vS32bBb/6hPp8EVeblGzMUhAyfc+RNbof
CERTIF:           IvUUlbp6CDy2qrCV4C0UHlyFMC4/11p2mZ63J0o88qL5sIAzwi7wuYfep+xjaYFv
CERTIF:           W8for8VZPtu9Fui6LyRvgv9OvUy5R+1wW2F1IChM9G4AbxdaYrtjtXuCuTVkfgdC
CERTIF:           CYabxcEQeh4uhjNqhP3hxAag7CI3KUud26w4Q/9kQ2BWxXMheZpDGkapfzJypbbe
CERTIF:           jLC7OHhqV6r+GoD8VmNMlKNQ/Jzvrf23Z2QLCuzSuIGwVIyKJG5y5ifecwARAQAB
CERTIF:           tCdKb25hdGhhbiBNaWNoYWxvbiA8am9obmRlc2NzQGdtYWlsLmmvbT6JAjcEEwEI
CERTIF:           ACEFAkvXHzcCGwMFCwkIBwMFFQoJCAsFFgIDAQACHgECF4AACgkQqS0qaGeNoc1Z
CERTIF:           EA//YQ13Sl5rxBQuXZcxWvsBlON4ZjqfZB6HnOdeeIdEnR7hEVgICDD/3AEZKXs5
CERTIF:           38uwTOlh++2T47KsbBaKFK+bCJfzivSVHj6SWWlvhJZo5WUcNZ6NxX5iBfM+IZ9r
CERTIF:           /2z8LS+t/iv6jy0tNeziVCxIDf3KJ9FIUl0QOrKRMIaj9q66rkVk9qZuNQRfx+3B
CERTIF:           9g1eKyhLj/CFjzz1yANIK7Ga74bgYhXTxHEGK+JKOf+tq4RymYpdRu87YauwL3ZY
CERTIF:           KbQbG4JzNWsqnJint7pueiEOrCcTHecUgkpQpPXrk/J2Xt07XudvteUcArOoRNP4
CERTIF:           qdb9ku1kkbtlsc74DQwE9sysbZbYoIT1YD1y0x2S9IqYsW/NpsOCek+8viRUSTHK
CERTIF:           mHnYcR2RDB/6EkkU2Rwjjt3c1idnrl2u40jTtU6ZZ/3Cbqjs/HEvf5EFacxx1JVA
CERTIF:           RQI0pTja56qn3sk85avJfM1I4P0swG9Ga+wnb/K1nyuIBQJZO3W+SfpJFOgzmXBu
CERTIF:           M1IiXIjdCbzhHTcrbc9xnvaqXCf5L13ZQy+CpEyvVpAWOTd4s6Ia9PldlPaRWwIG
CERTIF:           faZKVjCKLaDKle75vrP48pinCFu5JJ+6BrTVR0LxJRsd/XF8fbcOwk4TH1z2/bOt
CERTIF:           NlNldqHMx3RD/NZW+5nMae/MugVWZZUXJEremM3WpopvdciIRgQQEQgABgUCTL75
CERTIF:           gQAKCRD41bgH8Cma2QcnAKCWEllqpyiJn0Usj9iJa0xxsUKv2wCdFr7Buq1Ib+iu
CERTIF:           ufv5awslcQYCaNW0MEpvbmF0aGFuIE1pY2hhbG9uIChQZXJzbykgPGpvbmF0aGFu
CERTIF:           QG1pY2hhbG9uLmV1PokCNwQTAQgAIQUCUcNc4wIbAwULCQgHAwUVCgkICwUWAgMB
CERTIF:           AAIeAQIXgAAKCRCpLSpoZ42hzYaHEACqIwSOGZBPK6oT1S5PEweHC4MxgeaQQ1/0
```

```
CERTIF:          uF3u4G/hEo20zVzrv4PSBuRwdBfCjnZE3NakRfJGx5Y4T2dkYrZXNrB2DG+kqyqw
CERTIF:          KMa06gC4ZGK+DyVusU5+qCLDeTXvtY6W5z7GQo1SV6iHE0iFKqoqjuaXXG8Wuh3t
CERTIF:          C12VoQOxWUi6VE6Z8Ys+E76YpLztOhy4nFeEBL8w1NIq8KDKlydjbLnBkhUes0IP
CERTIF:          tIdLynxqm+tkRUA9WaPSIY/KQoFAbTLP3WAjnNM7wCqShAwgTBhH8v66pYKt/qpk
CERTIF:          jvVCEx3C48TTDgmHXlhrZTYR5jA4wpnrJwP+FhyBdA+fKPmJHzmsNwT321fH1Z40
CERTIF:          x+xfTZSVjfkgoW9cl95Sy3JGeDlomHjmR5cURz7Iwyg02z+k86Ve44/TKMXgXa2F
CERTIF:          H7D8Zjv9AYP4xKzDF+QnPrl55KzotMX9r501AaYB913jJ14xRMFjqWphB2hijrhQ
CERTIF:          hebW5a/HI+aKLyWkdNOA7N0AYhJ5MAjaKjsG7LfOo69X7HUtWtLeOATjJFT8CgFB
CERTIF:          FfVJBrK5GQ1v6l7dgUKMgUsKCAhKE7TbU6sqWTT+ohFTHcHhlziwWA+3jEfhhR2y
CERTIF:          JFUBhZ/nFo2v69Okav29LKTooHKaFpHzXLbvUb4P/G/wX15LffHHYb9JvU+PXP1r
CERTIF:          ddSIxmwKRw==
CERTIF:          =zj2g
CERTIF:          -----END PGP PUBLIC KEY BLOCK-----
org:             ORG-TSTA1-TEST
admin-c:         JKB1-TEST
tech-c:          JKB1-TEST
MNT-BY:          TESTASSO-MNT
CHANGED:         ripedb@example.com 20140426
SOURCE:          TEST
```

Ce qui nous donne l'objet suivant une fois le formulaire validé :

```
whois -h whois-test.ripe.net -B -r PGPKEY-698DB1DD
% Information related to 'PGPKEY-698DB1DD'

key-cert:       PGPKEY-698DB1DD
method:         PGP
owner:          Jean-Kevin Boulay <jkb@example.net>
fingerpr:       E99E BD56 AA68 C0CD FEC2  EB25 A92D 2A68 698D B1DD
certif:         -----BEGIN PGP PUBLIC KEY BLOCK-----
certif:         Version: GnuPG v1.4.12 (GNU/Linux)
certif:
certif:         mQINBEvXHzcBEACwopvTtQLlh+vfqwolOHaplhV8x1zFjdllT1E0CLwpsC5qpmQb
certif:         o5bNlSoDDqlMG6IHP98TUWM24mZU626MuWHte/vMOE5X1Q7tk0TRREeYjXxBftaa
certif:         9pGhynrjdk4FSDSZjZ5N0CgCDeX61j1TgCH0LT1EPB0/hJ+Tv2jDHc0vfCq6tnCg
certif:         a3GoKrAlPOWff/nNFpXQwpLbeFDizurGXNDpGa5yBPfn0k8bKzhPa+h9geYIx+4e
certif:         NbtG0J/g+GJuLhmlZyMVQ5+VYgeggbwLw8CvvcsH82v5PG5XIyceMqEXOvji5BOL
certif:         jzuYFYF/XinPbWgz5q2+5hV3HsJz8jMHlkPZnj1hbHa9mT1A+xSFUQ95gbvRXgF2
certif:         iOmewYUVPKoOfIhGis1cq1f4SlhrgO0vS32bBb/6hPp8EVeblGzMUhAyfc+RNbof
certif:         IvUUlbp6CDy2qrCV4C0UHlyFMC4/11p2mZ63J0o88qL5sIAzwi7wuYfep+xjaYFv
certif:         W8for8VZPtu9Fui6LyRvgv9OvUy5R+1wW2F1IChM9G4AbxdaYrtjtXuCuTVkfgdC
certif:         CYabxcEQeh4uhjNqhP3hxAag7CI3KUud26w4Q/9kQ2BWxXMheZpDGkapfzJypbbe
certif:         jLC7OHhqV6r+GoD8VmNMlKNQ/Jzvrf23Z2QLCuzSuIGwVIyKJG5y5ifecwARAQAB
certif:         tCdKb25hdGhhbiBNaWNoYWxvbiA8am9obmRlc2NzQGdtYWlsImmvbT6JAjcEEwEI
certif:         ACEFAkvXHzcCGwMFCwkIBwMFFQoJCAsFFgIDAQACHgECF4AACgkQqS0qaGeNoc1Z
certif:         EA//YQ13Sl5rxBQuXZcxWvsBlON4ZjqfZB6HnOdeeIdEnR7hEVgICDD/3AEZKXs5
certif:         38uwTO1h++2T47KsbBaKFK+bCJfzivSVHj6SWWlvhJZo5WUcNZ6NxX5iBfM+IZ9r
certif:         /2z8LS+t/iv6jy0tNeziVCxIDf3KJ9FIUl0QOrKRMIaj9q66rkVk9qZuNQRfx+3B
certif:         9g1eKyhLj/CFjzz1yANIK7Ga74bgYhXTxHEGK+JKOf+tq4RymYpdRu87YauwL3ZY
certif:         KbQbG4JzNWsqnJint7pueiEOrCcTHecUgkpQpPXrk/J2Xt07XudvteUcArOoRNP4
certif:         qdb9ku1kkbtlsc74DQwE9sysbZbYoIT1YD1y0x2S9IqYsW/NpsOCek+8viRUSTHK
certif:         mHnYcR2RDB/6EkkU2Rwjjt3c1idnrl2u40jTtU6ZZ/3Cbqjs/HEvf5EFacxx1JVA
certif:         RQI0pTja56qn3sk85avJfMlI4P0swG9Ga+wnb/K1nyuIBQJZO3W+SfpJFOgzmXBu
certif:         M1IiXIjdCbzhHTcrbc9xnvaqXCf5L13ZQy+CpEyvVpAWOTd4s6Ia9PldlPaRWwIG
certif:         faZKVjCKLaDKle75vrP48pinCFu5JJ+6BrTVR0LxJRsd/XF8fbcOwk4TH1z2/bOt
certif:         N1N1dqHMx3RD/NZW+5nMae/MugVWZZUXJEremM3WpopvdciIRgQQEQgABgUCTL75
certif:         gQAKCRD41bgH8Cma2QcnAKCWEllqpyiJn0Usj9iJa0xxsUKv2wCdFr7Buq1Ib+iu
certif:         ufv5awslcQYCaNW0MEpvbmF0aGFuIE1pY2hhbG9uIChQZXJzbykgPGpvbmF0aGFu
certif:         QG1pY2hhbG9uLmV1PokCNwQTAQgAIQUCUcNc4wIbAwULCQgHAwUVCgkICwUWAgMB
certif:         AAIeAQIXgAAKCRCpLSpoZ42hzYaHEACqIwSOGZBPK6oT1S5PEweHC4MxgeaQQl/0
certif:         uF3u4G/hEo20zVzrv4PSBuRwdBfCjnZE3NakRfJGx5Y4T2dkYrZXNrB2DG+kqyqw
certif:         KMa06gC4ZGK+DyVusU5+qCLDeTXvtY6W5z7GQo1SV6iHE0iFKqoqjuaXXG8Wuh3t
certif:         C12VoQOxWUi6VE6Z8Ys+E76YpLztOhy4nFeEBL8w1NIq8KDKlydjbLnBkhUes0IP
certif:         tIdLynxqm+tkRUA9WaPSIY/KQoFAbTLP3WAjnNM7wCqShAwgTBhH8v66pYKt/qpk
certif:         jvVCEx3C48TTDgmHXlhrZTYR5jA4wpnrJwP+FhyBdA+fKPmJHzmsNwT321fH1Z40
certif:         x+xfTZSVjfkgoW9cl95Sy3JGeDlomHjmR5cURz7Iwyg02z+k86Ve44/TKMXgXa2F
certif:         H7D8Zjv9AYP4xKzDF+QnPrl55KzotMX9r501AaYB913jJ14xRMFjqWphB2hijrhQ
certif:         hebW5a/HI+aKLyWkdNOA7N0AYhJ5MAjaKjsG7LfOo69X7HUtWtLeOATjJFT8CgFB
certif:         FfVJBrK5GQ1v6l7dgUKMgUsKCAhKE7TbU6sqWTT+ohFTHcHhlziwWA+3jEfhhR2y
certif:         JFUBhZ/nFo2v69Okav29LKTooHKaFpHzXLbvUb4P/G/wX15LffHHYb9JvU+PXP1r
certif:         ddSIxmwKRw==
certif:         =zj2g
certif:         -----END PGP PUBLIC KEY BLOCK-----
org:            ORG-TSTA1-TEST
admin-c:        JKB1-TEST
tech-c:         JKB1-TEST
mnt-by:         TESTASSO-MNT
changed:        ripedb@example.com 20140426
source:         TEST
```

La deuxième étape est de lier votre nouvel objet key-cert à votre objet de type mntner. Cela se fait avec un attribut « auth » dans l'objet mntner. Exemple : « auth: PGPKEY-698DB1DD ».

Vous pouvez désormais soumettre des objets signés par mail au robot du RIPE :

- Il faut obtenir une copie complète (-B) de l'objet que vous souhaitez modifier. Exemple :

```
whois -h whois-test.ripe.net -B -r JKB1-TEST > jkb1
```

- On modifie avec notre éditeur de texte favori. Dans mon cas, j'ajoute un attribut « remarks » et un attribut « changed ».

- On signe en gardant le texte en clair (on ajoute la signature à la fin, voir : Gnu Privacy Guard (GnuPG) Mini Howto (Français)) :

```
gpg -u <id_de_votre_cle> --armor --clearsign jkb1
```

On obtient un fichier jkb1.asc qui contient quelque chose dans ce genre :

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA256

Avertissement : des options RIPE ont été utilisées avec un serveur classique
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Information related to 'JKB1-TEST'

person:         J-K Boulay
address:        FR
phone:          +33123456789
e-mail:         ripedb@example.com
nic-hdl:        JKB1-TEST
mnt-by:         TESTASSO-MNT
remarks:        Ceci est un test
changed:        ripedb@example.com 20140418
changed:        jkb@example.net 20140426
source:         TEST

% This query was served by the RIPE Database Query Service version 1.72 (DBC


-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.12 (GNU/Linux)

iQIcBAEBCAAGBQJTXBLSAAoJEKktKmhnjaHNb4MP/0aF2GABfKvilKSrGiCn7RUH
yQ9k4D8SN7nJi6jJK9skvcELTp7eTEuCQmvNTuO2RFg5362bm1UNOcjKZep5++8h
i79wcjolfyxjkFrlOJaLVDmF1fVR+aqo5+ZoWNfbnba5pP8HyVY+n7027OvDO2yL
mwcGSKD7h/Za2dYl0wnWF9dlmcIGcYk+m5ruKb3bqjpTF79n4S3/ua+WzACkX9UH
Mg2B60xJoVIAO0A/o7y6UvxCO7+8mI/E3Y2B4AVYOvRaJ1QnJpU/Ty1csx9pDbJg
dicChmOaZ+EbXYHLi0axc8ixucpo0mFIXQ4nIV1lear3E5EaVv7wx/wr1r7d5BYi
g8BpLGB60hYBISpoEUlWF/V8hxf7pM3QNUagm+MjLzvMFLLcx9s58P0urVzijPPX
b1H4Tzorx6Lcm512Gcu2IfHGVVxO0JTfCXPgubDTV2QcSS4RL5vF7C8J0nrGr59m
Nw5hUBW+J0nV1418yVB4oXIC+Q0LFN5rfFRRlahhTZ5aGR/ieU+Ftk6MPE+Nt43o
bfO64IfCkrb4XKSFzwbpCcFlAkImFuun74L+dOfVstzEMYc0p5XetN1IEhPeppiv
j0S/p1tmCDHEkUZUkl51xAeRiQdqqVwx4P20tW7P9Dt7H4yqABgas6Yyia5DVjLj
MZYvC/9Fm4s1ysKQctwP
=74PE
-----END PGP SIGNATURE-----
```

- On envoie notre objet signé avec notre MUA préféré. Exemple :

```
mail -s "" test-dbm@ripe.net < jkb1.asc
```

On reçoit un mail récapitulatif : réussite ou échec, ce qui ne s'est pas bien passé, ... Exemple :

```
SUMMARY OF UPDATE:

Number of objects found:                  1
Number of objects processed successfully:  1
  Create:         0
  Modify:         1
  Delete:         0
  No Operation:   0
Number of objects processed with errors:   0
  Create:         0
  Modify:         0
  Delete:         0

DETAILED EXPLANATION:
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The following object(s) were processed SUCCESSFULLY:

---
Modify SUCCEEDED: [person] JKB1-TEST   JK Boulay



~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The following paragraph(s) do not look like objects
and were NOT PROCESSED:

Avertissement : des options RIPE ont été utilisées avec un serveur classique.
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Information related to 'JKB1-TEST'

% This query was served by the RIPE Database Query Service version 1.72 (DBC-WHOIS4)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

The RIPE Database is subject to Terms and Conditions:

http://www.ripe.net/db/support/db-terms-conditions.pdf


For assistance or clarification please contact:
RIPE Database Administration <ripe-dbm@ripe.net>

Generated by RIPE WHOIS Update version 1.72 on DBC-WHOIS2
Handled email update (TEST, 2014-04-26 23:42:04)
```

Il est possible de soumettre plusieurs objets en même temps, signés (avec la même clé ou non) ou non-signés (il faudra alors un mot de passe dans un attribut « password ») ce qui est très pratique pour une modification de masse. Un maintainer peut avoir plusieurs clés PGP. L'authentification se fait "en OU LOGIQUE" sur tous les moyens d'authentification (clé PGP ou non) associés à un des objets maintainer qui sont associés à l'objet à modifier.

Merci à Johndescs pour ces tests.
Fin de l'édit

Faire le ménage dans la base de test

Pour supprimer tous les objets que vous avez créés dans la base de test, il faut les rechercher et choisir l'option « delete ». À cause de la référence circulaire entre vos objets de types mntner et person, vous ne pourrez pas les supprimer directement. Néanmoins, sur cette base de test, on peut ruser : casser la référence circulaire en définissant « TEST-DBM-MNT » comme mntner de l'objet personn et « AA1-TEST » comme admin-c de l'objet mntner. Ça fonctionne uniquement car nous avons le mot de passe de « TEST-DBM-MNT ». J'imagine qu'il existe une méthode plus pragmatique voire un bête formulaire web mais je ne l'ai pas trouvé. Notons quand même que cette base de données est purgée tous les jours à 0h.

Dernières remarques

On notera que les attributs « tech-c » et « admin-c » de tous les objets peuvent pointer sur un seul et même objet de type person.

Enfin, si l'on souhaite visualiser les objets de la base du RIPE sous un format graphique conviviale, l'onglet « Database » de l'outil Ripestat est là pour ça.

ÉDIT du 26/04/2014 à 22h00 :
L'attribut « changed » doit dire qui a modifié/créé un objet et quand. Qui est une adresse email. Quand est une date au format AAAAMMJJ. Il n'est pas nécessaire de remplir la date, elle est auto-complétée. L'attribut changed ne se met pas à jour tout seul lors d'une modification. Il peut y avoir plusieurs attributs changed pour un même objet, ce qui permet d'avoir un historique des modifications de cet objet.

Ne pas créer/modifier deux objets en parallèle dans deux onglets différents du même navigateur si les deux objets ne sont pas dans la même source (RIPE/TEST) sinon l'application web se prend les pieds dans le tapis et vous hurle dessus : « Unrecognized source: TEST ». Dans cet exemple, ça parle de TEST alors que

je voulais soumettre un objet dans RIPE mais comme j'avais ouvert ce formulaire puis ouvert un autre onglet dans TEST pour faire un test ...

Ce formulaire permet d'avoir rapidement l'adresse mail à contacter en cas d'abus. Il suffit de saisir une adresse IP, un ASN ou un inet(6)num. Cela retourne la valeur de l'attribut « abuse-mailbox » d'un objet de type « role » (qui représente une personne/un groupe de personnes dédié à la même tâche (administratif, adminsys, ...) au sein d'une organisation) lui même associé à un objet de type « organisation » (avec un attribut « abuse-c ») lui même associé à la ressource recherchée (aut-num ou inet(6)num). Si cette recherche est infructueuse (car pas d'objet organisation, car pas d'attribut « abuse-c », ...), elle se poursuit récursivement sur les ressources englobantes (as-block englobant, inet(6)num englobant). Merci à Lulu pour la découverte de ce formulaire.
Fin de l'édit

Catégorie: Administration réseau, Divers / Tags: aucun tag / Ajouter un commentaire

Aucun commentaire. Ajoutez votre commentaire

Votre commentaire

Nom* : Email* : URI :

Soumettre

Le contenu de ce blog est mis à disposition sous un contrat Creative Commons BY-SA 3.0 France. Exception faite des commentaires, des extraits de textes, médias et marques qui restent la propriété de leurs auteurs respectifs. Toute copie de ce site doit mentionner http://www.guiguishow.info. Publié avec WordPress. Pyrmont V2